```
Minu
-------
    1.

Shankar
-------
    1. Divide without if.
    2. Top ranked words.

Ayush
-------
    1. problem in MapEntry
    2. dot product, how.


Saurab
-------
    1. function composition problem.

==========================================
================
Statements based language
    * C, C++, Java, Python...
    for(expr1;expr2;expr3){

    }


Expressons based language
    * Scala, Clojure

    (defn -main ...)
    {2 + 3 * () + ....}

    (+ 2 3 ...)
    Precedence, evaulates

    Scala => def f1 = if(expr) arg1 else arg2
    Clojure => (defn .. if cond arg1 arg2)

=================
Statements based langs  ============> mid point <========= expr lgs

=================

partially defined function to total function =>

=================
(key, val) tuple

(into {} ...seq of tuple)

------------------

V1 => {2 4, 5 1, 9 100} [0 0 4 0 0 1 0 0 0 100]
V2 => {5 3, 9 40, 100 4, 300 8}

5000 ngrams => dimensions 5000.

iterate 1 to 50000.

I dont want to go that far.

Accumulates: keys #{2, 5, 9} #{5, 9, 100, 300}
```

```

Union of two sets, converts to vec, sort.

[2, 9, 9, 100, 300] => iterate to these 5 values.

(get v1 2) * (get v2 2)
...
reduce to +.

=======================
Relevant Words Optimization
===========================

1000 words

i_word => iterate to 1000 words

Inverted index:

player => {pla, lay, aye, yer, play, laye, ayer}

pla => [play, plays, plan]
lay => []

==============================

[[play, plays, plan] [] []]
flatten, unique

100 words.

==============================

function composition
--------------------
f(x) => x^....
g(x) =>

f(g(x)) =>
g(f(x)) =>

f(g(x)) != g(f(x))
--------------------
TF/IDF metric: term frequency (no. of appear) * inverse document frequency (no of word that gram presence)
--------------------

System Evaluation
------------
Precision
Recall
FMeasure
------------
thesis: google (santa basnet theis)
------------------------------------
```